PAIRWISE TAIL-SPAN AUTHORIZATION FOR RESOURCE-CONSTRAINED FILTERED SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Across a 50M-vector filtered-retrieval stress grid, the fastest recall-feasible action changes on 364/940 adjacent edges in the 5–15% selectivity band, versus 351/3,860 edges elsewhere. These reversals turn a plausible top-ranked plan into lost generation slack when its fanout or SSD traffic produces a latency-tail error. RAVEL makes execution of that winner a pair-conditioned authorization decision: held-out recall tables and RAM, SSD, and latency caps define feasible P1–P4 actions; calibrated median-to-P95 spans characterize the leading pair; and ambiguous requests take a regime-local fallback. On dual AMD EPYC 7543 CPUs with 128GB RAM and a 1TB SSD budget, RAVEL reaches 37ms P95 and 56ms P99 at a common 0.95 Recall@20 target, improving on GateANN's 44/70ms by 16%/20% and Qdrant's 68/101ms by 46%/45%. With actions, requests, recall tables, provisional ranks, and fallback implementation fixed, the executed-policy mis-pick rate falls from 12.6% for winner-only dispatch to 4.9%. At 80% authorization coverage, RAVEL reduces selective P95 excess latency from RCPS-Tail's 14.6ms [12.8,16.4] to 9.6ms [8.3,10.9]. The enterprise-selected gate also reaches 38.9ms P95 on 356 public band queries, compared with 41.8ms for RCPS-Tail and 58.7ms for the native policy. Layering this measurable tail-risk decision over existing kernels preserves retrieval latency budget for downstream generation without requiring an index replacement.

1 INTRODUCTION

A filtered-search planner can satisfy its recall target, rank four tuned actions, and still spend the latency reserve needed by downstream generation. On 812 template-held-out Medium+EntDoc requests, unconditional execution of the predicted winner produces 47/77ms P95/P99 and mis-picks the fastest resource-feasible action on 12.6% of requests. A pair-conditioned gate over the identical action menu and fallback implementation reaches 37/56ms and a 4.9% executed-policy mis-pick rate. The missing planner output is therefore not another top-1 label, but evidence that the proposed winner is separated enough from its nearest feasible alternative to execute.

This decision failure has a measurable geometry. Predicate selectivity controls how many items survive a filter; vector–metadata correlation controls where those survivors appear during ANN traversal. A 50×50 training-side stress grid spans 0–50% selectivity and correlation from 0 to 1. Among horizontal and vertical edges whose endpoints remain in the same region, the fastest recall-feasible action among filter-first (P1), ANN-first (P2), interleaved (P3), and partition-aware search (P4) changes on 364/940 edges in the frozen 5–15% band, compared with 351/3,860 edges elsewhere. The corresponding final-test band contains 812 of 3,248 enterprise requests.

Figure 1 encodes each cell by the P95 ratio between its slowest and fastest eligible actions and locates the reversal-heavy band. For a RAG service with end-to-end deadline τ, retrieval consumes generation slack whenever $L_{\mathrm{retr}} > \tau - L_{\mathrm{LLM}}$. Two predicates with equal selectivity can demand opposite plans because one aligns with embedding neighborhoods while the other sends expansion into cold SSD regions. The same resource coupling appears in RAG (Lewis et al., 2020; Gao et al., 2023) and DPR- or ColBERT-based pipelines (Karpukhin et al., 2020; Khattab & Zaharia, 2020): a tail error delays evidence even when the index and retriever stay fixed.

RAVEL makes the leading feasible pair the authorization object. Held-out recall tables first choose quality-feasible fanout; latency, RAM, and SSD constraints then remove infeasible actions. Quantile

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

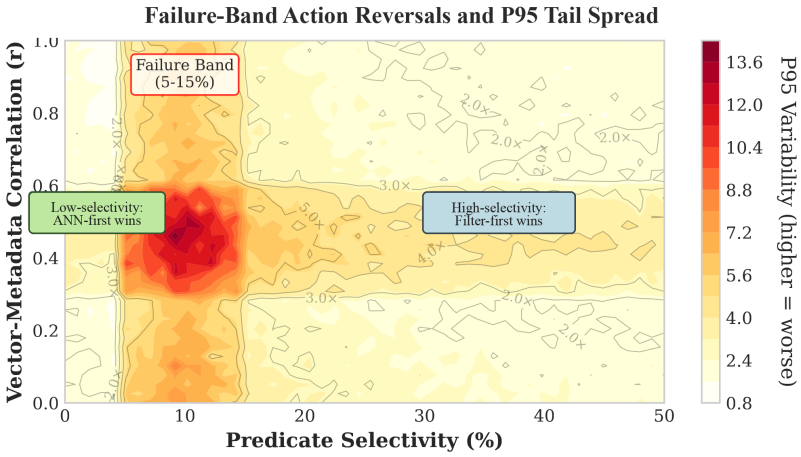


Figure 1: P95 max/min action ratio across the 50×50 Medium+EntDoc stress grid. The 5–15% boundary is selected on training-side traces and frozen before final testing. Among horizontal and vertical edges whose endpoints remain in one region, the paired-trace argmin changes on 364/940 in-band edges (38.7%) and 351/3,860 outside edges (9.1%); exact ties use plan ID.

models predict a median-to-P95 span for each survivor. RAVEL executes the provisional lowest-tail action when its normalized overlap with the runner-up is small and aggregate uncertainty is bounded; otherwise a recall-eligible regime-local action executes. Authorization coverage, executed-policy mis-picks, and selective P95 excess latency expose both the effect and the cost of vetoes.

This decision object separates RAVEL from three neighboring approaches. Learned filtered-ANN planning (Gan & Wang, 2026) ranks actions, GateANN (Lee et al., 2026) changes SSD traversal and layout, and selectivity/correlation studies (Amanbayev et al., 2026) characterize operating regions. RAVEL leaves P1–P4 and their indexes intact and tests the predicted leading pair after recall and resource filtering. RCPS-Tail provides the strongest statistical comparator by calibrating one-sided excess latency; matched-coverage sweeps then distinguish pair-conditioned selection from merely vetoing more requests.

The closest-system comparison favors the gate despite GateANN's lower SSD-read count: at a common 0.95 Recall@20 target, RAVEL reaches 37/56ms P95/P99 versus GateANN's 44/70ms. At 80% authorization coverage, selective P95 excess is 9.6ms [8.3,10.9], versus 14.6ms [12.8,16.4] for RCPS-Tail. On 356 public band queries, the enterprise-selected thresholds retain a 38.9ms P95 point estimate, and outside the band RAVEL remains within 2.8ms of the best public-workload policy.

Contributions. The paper makes three connected advances: (i) an edge-counted selectivity–correlation protocol that identifies a recurring plan-reversal band; (ii) a planner–engine authorization rule that combines an explicit resource-feasible set, leading-pair tail-span overlap, and regime-local fallback (Algorithm 1); and (iii) an identical-action, equal-coverage evaluation that compares winner-only, point-margin, width-only, RCPS-Tail, and pair-conditioned controllers before testing common-recall-target systems, public queries, transfer, and scale (Figure 3; Tables 1–3).

The evaluated envelope spans 10M–100M single-index settings and a four-node service with four disjoint 50M tenant shards (200M aggregate indexed capacity); every request searches one shard. Across this CPU-only envelope, pair-conditioned authorization improves filtered retrieval under explicit recall, latency, RAM, and SSD constraints.

2 Related Work and Positioning

Filtered search exposes three distinct decisions: how an index executes a predicate, which available action ranks first, and whether that apparent winner merits execution. HNSW and product quantization (Malkov & Yashunin, 2020; Jégou et al., 2011), SPANN (Chen et al., 2021), Filtered-DiskANN (Gollapudi et al., 2023), UNIFY (Liang et al., 2024), ACORN (Patel et al., 2024),

SIEVE (Li et al., 2025), SeRF (Zuo et al., 2024), tag-filtered ANN (Luo et al., 2025), joint label–vector indexing (Cai et al., 2024), and dynamic range-filtered indexing (Zhang et al., 2025a) address the first decision through graph traversal, partitioning, or predicate access. GateANN (Lee et al., 2026) specializes that layer for SSD traffic. The learned planner of Gan & Wang (2026) addresses the second decision, while Amanbayev et al. (2026) map the selectivity–correlation geometry in which rankings change. RAVEL addresses the third: after recall and resource feasibility are fixed, it tests whether the actual leading pair is separated enough to execute the winner.

Selective prediction and learned query optimization connect predictions to uncertainty, calibration, and fallback (Marcus et al., 2021; Sun et al., 2021; Kim et al., 2022; Thirumuruganathan et al., 2022; Zhang et al., 2023; Wang et al., 2023; Zheng et al., 2023; Zhang et al., 2024; Heinrich et al., 2025b;a; Zhang et al., 2025b; Liu et al., 2025). RCPS-Tail is the strongest same-feature methodological comparator in this paper: it applies a one-sided split-conformal bound directly to excess latency and uses the same regime-local fallback. RAVEL instead scores normalized overlap between the calibrated median-to-P95 spans of the two leading feasible actions. The score is an empirical pairwise decision statistic rather than a conformal interval for their latency difference. This distinction is operational: a marginal width can be small while the runner-up occupies the same tail region as the proposed winner.

Table 2 compares authorization with learned planning and SSD-native execution; Table 1 removes backend and action-menu differences; and Table 3 applies the frozen gate thresholds to public queries. HAKES (Hu et al., 2025), HARMONY (Xu et al., 2025), GaussDB-Vector (Sun et al., 2025), and FedVSE (Fan et al., 2025) broaden the deployment envelope surveyed by Lin et al. (2025). Their storage and traversal choices determine the candidate tails; pair-conditioned authorization consumes those tails at the planner–engine boundary.

3 Pair-Conditioned Tail-Span Commitment

RAVEL intervenes between a filtered-search planner and its engine. Recall eligibility determines which actions can satisfy quality, budget filtering removes actions outside the request envelope, and a pair-conditioned score decides whether the provisional winner executes. A veto routes the request to a recall-eligible regime-local action. The resulting record exposes the feasible set, competing pair, gate state, and fallback choice for each request.

3.1 Requests, Feasibility, and Architecture

We model a request and its active deployment envelope as $Q = (q, \Pi, k, \tau, m_{\max}, d_{\max})$, where q is the embedding, Π is a conjunction of metadata predicates, k is the requested result count, τ is the latency target, and $(m_{\max}, d_{\max})$ cap aggregate RAM and SSD allocation for the routed shard. Each plan family maintains a held-out recall table indexed by dataset, hardware profile, workload family, requested k, and selectivity/correlation bin. For candidate (p, k'), $\widehat{R}$ is the empirical recall stored in this table, rather than an online learned estimate; sparse cells back off to the next coarser key. The controller selects among $p \in \{\mathrm{P1}, \mathrm{P2}, \mathrm{P3}, \mathrm{P4}\}$ and fanout k' through

$$
\begin{aligned}
(p^\star, k'^\star) &= \arg\min_{p,k'} \widehat{L}_{95}(p, k' \mid x) \\
\text{s.t.} \quad & \widehat{R}(p, k' \mid x) \geq \rho, \qquad \widehat{M}(p, k' \mid x) \leq m_{\max}, \\
& \widehat{D}(p, k' \mid x) \leq d_{\max}, \qquad \widehat{L}_{95}(p, k' \mid x) \leq \tau.
\end{aligned} \tag{1}
$$

Here x contains selectivity, correlation, cache, and I/O features. $\widehat{M}$ includes the routed shard's resident hot tier plus peak query workspace, while $\widehat{D}$ is the cold-index footprint under that tier profile. Let $\mathcal{F}(x)$ denote the candidates satisfying all four constraints. The recall tables and model checkpoint are frozen before each reported run. Equation 1 defines eligibility and a provisional optimum; authorization is a second decision.

Figure 2 shows the closed loop. The frontend normalizes the request, the recall table fixes feasible fanout, the estimator predicts tail spans and memory, and the planner applies Equation 1 before comparing the leading pair. The engine executes an unchanged search kernel over RAM and SSD tiers and returns telemetry. Appendix Table 4 specifies the interfaces and fallback state.

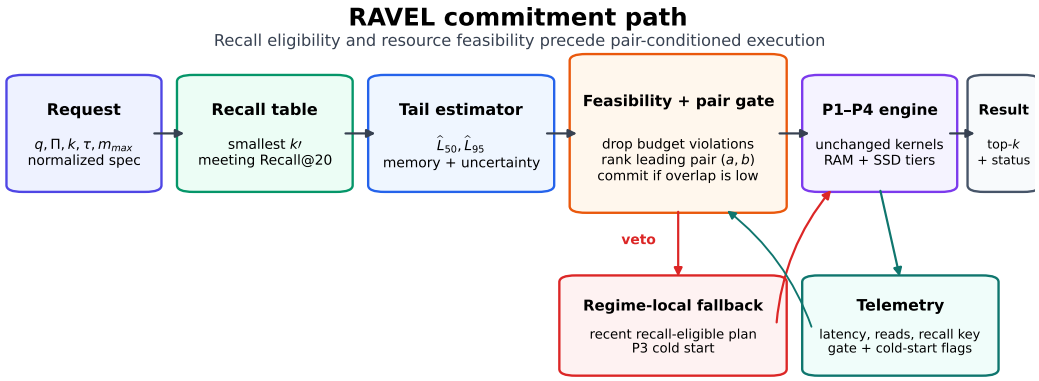


Figure 2: RAVEL authorization path. Recall eligibility and RAM, SSD, and latency feasibility precede the leading-pair overlap test. A veto uses regime-local fallback (P3 at cold start), while execution telemetry records the recall key, resource use, and gate state.

3.2 Actions and Workload Coordinates

RAVEL exposes four established strategies: P1 filters candidates before vector search; P2 expands ANN candidates before applying the predicate; P3 interleaves graph traversal and streaming predicate checks; and P4 prunes through predicate-aligned partitions before local ANN search. P4 is particularly effective for taxonomies such as `region/department/team` that align with IVF cells or DiskANN neighborhoods. Appendix Algorithm 2 specifies the P3 execution kernel.

The planner estimates two workload coordinates. For $\Pi = \bigwedge_j P_j$, selectivity is

$$\hat{s}_\Pi = \prod_j \hat{s}_{P_j} \cdot \prod_{(i,j)\in\mathcal{E}} \hat{\omega}_{P_i,P_j},$$

where marginal sketch estimates $\hat{s}_{P_j}$ are corrected by sampled pairwise factors $\hat{\omega}_{P_i,P_j}$. Before splitting requests, the estimator freezes a template-independent, metadata-stratified in-RAM landmark reservoir of 500 corpus IDs and embeddings. For each query, exact landmark similarities define rank score u_i, predicate evaluation defines $y_i = \mathbf{1}[\Pi(c_i)]$, and $\hat{r} = \mathrm{Spearman}(y, u)$; unseen templates therefore require neither history nor warm-up. Landmark scoring, predicate lookup, and rank reduction are included in the reported 0.18ms overhead. This feature reservoir is distinct from action fanout k'_p, and recall evaluation uses an exact brute-force oracle.

3.3 Selective Tail Risk and Pairwise Separation

Let $g_i \in \{0, 1\}$ indicate whether request i authorizes its provisional winner p_i, let f_i be the fallback selected after a veto, and let $\widetilde{p}_i = g_i p_i + (1-g_i) f_i$ denote the executed policy in plan-index notation. Offline evaluation replays every action in the same resource-feasible set $\mathcal{F}_i$ under the request's run state. The paired-trace oracle is

$$p_i^\dagger = \arg\min_{p\in\mathcal{F}_i} \ell_i(p),$$

with exact ties broken by plan identifier; it never enters the online gate. For N request–run observations with nonempty feasible sets, authorization coverage is $C(g) = N^{-1}\sum_i g_i$. We distinguish error among authorized provisional winners,

$$E_{\mathrm{auth}}(g) = \frac{\sum_i g_i \mathbf{1}[p_i \neq p_i^\dagger]}{\max(1, \sum_i g_i)},$$

from executed-policy error,

$$E_{\mathrm{exec}}(g) = \frac{1}{N}\sum_i \mathbf{1}[\widetilde{p}_i \neq p_i^\dagger].$$

Tables 1 and 7 report E_{exec}; selective tail risk is the 0.95 quantile of $\ell_i(p_i) - \ell_i(p_i^\dagger)$ among authorized requests. End-to-end P95/P99 includes fallback execution after vetoes. Singleton feasible sets execute their sole member and count as authorized; typed infeasible responses are counted separately and do not occur in the 812-query headline slice. Fallback percentages use all N query–run observations as their denominator.

For each feasible (p, k'), companion LightGBM quantile regressors produce a calibrated tail-quantile span

$$J_{p,k'} = [\widehat{L}_{50}(p, k' \mid x), \widehat{L}_{95}(p, k' \mid x)].$$

This span summarizes median-to-tail dispersion; it is not a classical two-sided prediction interval. A template-disjoint quantile-calibration partition adjusts the upper quantile toward empirical 0.95 coverage. Appendix Table 6 evaluates the frozen calibration on the final-test requests rather than reusing those requests to fit the adjustment. Plan uncertainty is

$$\hat{u}_{p,k'} = \frac{\widehat{L}_{95}(p, k' \mid x) - \widehat{L}_{50}(p, k' \mid x)}{\max(\widehat{L}_{95}(p, k' \mid x), 1\,\text{ms})}, \qquad \hat{u} = \max_{(p,k') \in \mathcal{F}(x)} \hat{u}_{p,k'}.$$

This feasible-set maximum is a global uncertainty guardrail; the overlap statistic below measures the decision geometry of the actual leading pair. For the two lowest-tail candidates a, b after feasibility filtering, the separation signal is the normalized overlap

$$\text{ov}(a, b) = \frac{\max\{0, \min(h_a, h_b) - \max(\ell_a, \ell_b)\}}{\min\{h_a - \ell_a, h_b - \ell_b\} + \epsilon},$$

where $J_a = [\ell_a, h_a]$ and $J_b = [\ell_b, h_b]$. Zero denotes disjoint spans; values near one indicate that the narrower span is almost contained in the other. Normalization by the smaller span makes the test scale-free, while conditioning on (a, b) measures the geometry of the actual execution choice. The statistic is an empirical authorization score: its value is tested by the matched-coverage reduction in executed-policy mis-picks and authorized-request excess latency in Figure 3.

Our strongest generic baseline, RCPS-Tail, fits a separate LightGBM predictor for paired-trace excess latency $z_i = \ell_i(a_i) - \ell_i(p_i^\dagger)$ using the same non-gate features. On a template-disjoint calibration partition, it forms $U_i = \widehat{z}_i + q_{0.95}(z - \widehat{z})$ with the finite-sample order statistic $\lceil 0.95(n_{\text{cal}} + 1) \rceil$. A request is authorized when $U_i \le \beta$; sweeping β and selecting matched-coverage points yields a one-sided split-conformal controller for the same tail loss and final-test protocol. RCPS-Tail, point-margin, width-only, and RAVEL share the feasible actions and fallback implementation, providing same-feature tests of the authorization score.

Algorithm 1 RAVEL pairwise tail-separation gate and telemetry update.

Require: request Q, features x, recall floor ρ, thresholds $(\theta, \sigma_{\max})$
for each $p \in \{\text{P1}, \ldots, \text{P4}\}$ **do**
  look up the smallest held-out-calibrated k'_p meeting ρ
  score (p, k'_p) for P50, P95, RAM, SSD, and latency feasibility
end for
discard infeasible candidates; return a typed infeasible response if none remain
if $|\mathcal{F}(x)| = 1$ **then**
  execute the sole feasible candidate and record a singleton decision
  return results and telemetry
end if
rank feasible candidates and set (a, b) to the lowest-P95 pair
if $\text{ov}(a, b) \le \theta$ **and** $\hat{u} \le \sigma_{\max}$ **then**
  authorize and execute a
else
  veto a; execute the safest recent feasible action, or the cold-start rule
end if
record action, fanout, latency, recall-table key, SSD reads, and gate outcome
return results and telemetry

Runtime features include $(\hat{s}, \hat{r})$, fanout, requested k, action identity, cache-miss probability, SSD queue depth, and recent fallback activity. A winner-only policy dispatches a unconditionally. On a

template-disjoint gate-validation partition, we select θ=0.15 and $\sigma_{\max}$=0.3 and freeze both before final testing. RAVEL authorizes a when $\mathrm{ov}(a, b) \leq \theta$ and $\hat{u} \leq \sigma_{\max}$. Otherwise, recent feasible actions for the same regime key—dataset, hardware profile, requested k, and selectivity/correlation bins—are ordered by recall eligibility, realized P95, and SSD reads, with ties favoring P3. At cold start, P3 is used when feasible; otherwise the lowest predicted-P95 feasible action is returned with a cold-start flag. Table 1 and Figure 3 compare authorization scores while holding these fallback rules fixed. Appendix Table 7 varies already-selected thresholds as a robustness analysis.

3.4 Tiered Storage and Resource Accounting

RAVEL executes over a two-tier index because the budget bottleneck includes resident memory, query workspace, SSD capacity, and random I/O. The hot tier stores frequently traversed HNSW neighborhoods (Malkov & Yashunin, 2020) in RAM; the cold tier uses PQ compression or SPANN-like partitioning (Jégou et al., 2011; Chen et al., 2021) on SSD. Equation 1 admits only tier profiles whose aggregate footprints fit $(m_{\max}, d_{\max})$, and the latency model prices hot-tier hit probability and cold-tier probes for every action.

In our feasibility study, an 85M-vector deployment of the two-tier design fits within a RAM footprint of 94GB and an SSD footprint of 420GB; an all-HNSW design with similar recall uses 153GB. This capacity-oriented point uses a colder compressed split than the latency-oriented scale configurations in Appendix Table 15. The planner prices both allocations as explicit constraints, converting the same control rule between capacity- and latency-oriented operating points.

Planner telemetry guides migration between tiers, with rate limits imposed to keep query latency stable. Vectors recurring in P3 frontiers also receive higher hotness from the planner, making tiering an optimizer component rather than solely a storage policy.

P3 evaluates the predicate before exact-distance insertion but does not stop graph expansion merely because the current node fails the predicate. Backend-native frontier ordering and the explicit fanout, time, and memory budgets terminate traversal. If fewer than k matches are found, the engine returns the available matches with an incomplete-result flag; that request remains in recall accounting and its telemetry can trigger safe mode. Across the evaluated regimes, the gap to exact brute-force Recall@20 stays below 0.02 once $k' \geq 100$; §4.5 reports the fixed-budget quality view.

3.5 Telemetry, Calibration, and Failure Handling

A compact telemetry record is logged for every query: its selected plan, $(\hat{s}, \hat{r}, \hat{u})$, a breakdown of latency, events in the cache, statistics of the frontier, and decisions about fallback. Online calibration aggregates these records over 5-second windows. If the latest 10K queries show a difference exceeding 15% between predicted and observed P95, the system retrains the cost model using the latest 50K telemetry records, while keeping the previous checkpoint available for rollback. In the reported headline experiments, these live retraining hooks are disabled throughout each run; Tables 1 and 8 therefore evaluate a fixed checkpoint, keeping deployment adaptation separate from evaluation.

Safe mode uses only executed-request telemetry. For chosen action p_i, the observable residual violation is $\mathbf{1}[\ell_i(p_i) > \widehat{L}_{95}(p_i \mid x_i)]$ after held-out calibration. If the one-minute residual-violation rate exceeds 15%, failure-band traffic temporarily uses P3 and other traffic uses threshold heuristics; normal authorization resumes after the rate falls below 5%. Sensitivity across 10%/3%, 15%/5%, and 20%/8% trigger pairs preserves the central tail advantage (Appendix Table 7). Mis-pick remains an offline evaluation metric computed from paired action traces; it is never required by the production trigger. Asynchronous retraining, rollback, and admission control keep model updates outside request execution.

4 Evaluation

The central experiment holds search actions, provisional ranks, and fallback fixed and asks whether pair-conditioned authorization improves the executed choice at equal coverage. We then connect that result to the reversal geometry, common-recall-target systems, public queries, calibration, and resource scaling.

4.1 SETUP AND PROTOCOL

Evaluation chronology. The enterprise trace contains 15K requests from three RAG deployments over seven days. Normalized predicate templates are grouped before any plan traces are expanded and assigned once to mutually exclusive model-training, recall-table, quantile-calibration, gate-validation, and final-test roles. The 5–15% boundary is selected only from training-side stress traces; quantile corrections are fit only on calibration templates; and $(\theta, \sigma_{\max})$ are selected only on gate-validation templates. The final test contains 3,248 requests, including 812 in the frozen band, and is opened after the recall tables, model checkpoint, calibration adjustment, and gate thresholds are fixed. Appendix Table 6 consequently audits one frozen provisional-action quantile per final-test request.

The public study applies the same template grouping and frozen 5–15% boundary to 1024 final-test MoReVec-style queries (Amanbayev et al., 2026). Workload-specific quantile models and RCPS-Tail corrections use a separate public calibration partition; RAVEL transfers the enterprise-selected overlap and uncertainty thresholds unchanged. Appendix Table 12 reports every final-test selectivity/correlation denominator.

Systems and data. We evaluate three CPU-only latency configurations. Small uses two Intel Xeon Gold 6248R processors with 64GB RAM; Medium uses two AMD EPYC 7543 processors with 128GB; Cluster uses four 24-core, 48GB nodes connected by 25GbE. Every machine runs Ubuntu 22.04 on Samsung PM9A3 NVMe SSDs with pinned threads, NUMA-aware allocation, a predicate JIT, and GPUs disabled. EntDoc contains 50M 768-dimensional vectors, EcomProd 100M 512-dimensional vectors, and LegalDoc 20M 1024-dimensional vectors. The primary operating envelope is 0.1–30% selectivity and Spearman correlation from 0.08 to 0.52; the stress grid and descriptive trace-tail bins extend to 50% and 1.0 to expose transitions outside that concentration. The scale matrix uses a document-ID-stable 10M LegalDoc subset and four disjoint 50M EntDoc tenant shards for 200M aggregate service capacity, with one shard searched per request. The separate capacity sweep in Figure 5 uses an additional 256GB host with up to 224GB allocatable RAM.

Two non-interchangeable protocols. The primary latency protocol sets the recall-table floor to ρ=0.95 Recall@20 against exact brute force and gives each method its smallest eligible fanout. Discrete fanout and table-to-final transfer produce the small achieved-recall differences reported beside every latency result. Appendix Table 11 instead fixes an identical 80ms deadline and permits recall to vary. The template-independent 500-landmark reservoir is frozen before splitting, and its per-query scoring is included in controller overhead. Each point uses a 30s warm-up, page-cache flushes, a fixed checkpoint, 5 independent runs, and 95% intervals from 1000 run-level bootstrap resamples. Controller comparisons replay the same request order from the same pre-run fallback-cache snapshot.

Baselines include filter-first, ANN-first, Milvus v2.3.4, Qdrant v1.7.3, ACORN (Patel et al., 2024), SIEVE (Li et al., 2025), the learned planner of Gan & Wang (2026), and GateANN (Lee et al., 2026). HNSW systems use M=32, ef_c=200, and ef_s=128; IVF systems match list count and probing depth before narrow-neighborhood tuning at the common recall target. Appendix Table 10 records the decision-critical settings.

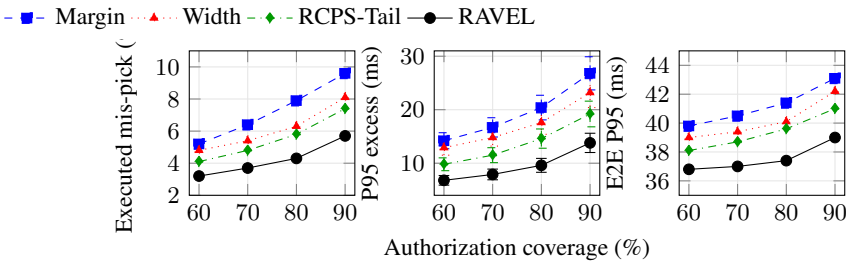
4.2 DOES PAIR CONDITIONING IMPROVE THE SAME ACTION MENU?

Table 1 fixes hardware, requests, recall tables, P1–P4 kernels, training data, provisional ranking, replay order, and fallback implementation. Winner-only execution produces 47/77ms tails and a 12.6% executed-policy mis-pick rate. Pair-conditioned authorization reaches 37/56ms, 4.9%, and 618 SSD reads. RCPS-Tail reaches 39/61ms and 5.8%. The equal-coverage selective-risk analysis below supplies the direct comparison between their authorization scores.

Threshold sweeps compare every selective controller at the same authorization coverage. At 80%, RAVEL records 4.3% executed-policy mis-picks, 9.6ms selective P95 excess [8.3,10.9], and 37.4ms end-to-end P95. RCPS-Tail records 5.8%, 14.6ms [12.8,16.4], and 39.6ms. Equal coverage and a common fallback rule rule out veto frequency as the explanation for the separated excess-latency intervals.

Table 1: Controller isolation on 812 final-test Medium+EntDoc band queries (4,060 query–run observations). Rows share P1–P4, recall tables, budgets, provisional ranks, replay order, and the regime-local fallback implementation; only the authorization score changes. Mis-pick compares the executed action, including fallback, with the resource-feasible paired-trace oracle. Fallback uses all 4,060 observations as denominator. Brackets are marginal 95% run-level bootstrap CIs.

Policy	P95 [CI]	P99 [CI]	R@20	Exec. mis-pick	Fallback	Reads
Winner-only	47 [44.8,49.5]	77 [72.1,82.9]	0.956	12.6%	0	932
Point margin	42 [40.0,44.2]	67 [63.1,71.9]	0.957	8.8%	12.1%	761
Width-only	40 [38.2,42.1]	63 [59.3,67.8]	0.957	7.0%	15.2%	689
RCPS-Tail	39 [37.2,41.1]	61 [57.5,65.3]	0.958	5.8%	16.9%	655
RAVEL	37 [35.9,38.4]	56 [53.4,59.6]	0.958	4.9%	18.3%	618



Policy at 80% coverage	Exec. mis-pick	Sel. tail risk [95% CI]	P95 (ms)
Point margin	7.9%	20.4 [18.1,22.7]	41.4
Width-only	6.3%	17.6 [15.4,19.8]	40.1
RCPS-Tail	5.8%	14.6 [12.8,16.4]	39.6
RAVEL	4.3%	9.6 [8.3,10.9]	37.4

Figure 3: Selective-risk frontiers on final-test Medium+EntDoc band queries. The left and right panels include fallback outcomes; the middle panel reports P95 excess latency among authorized provisional winners. Error bars are 95% run-level bootstrap intervals. At equal coverage, pair-conditioned authorization yields the lowest measured executed-policy mis-pick rate, selective excess latency, and end-to-end P95.

4.3 Where Does Authorization Matter Most?

On Medium+EntDoc, the fastest recall-feasible action changes on 364/940 adjacent edges in the frozen 5–15% band and 351/3,860 edges elsewhere. Within the band, RAVEL reaches 37ms P95 and 56ms P99 at the common recall target. GateANN, the nearest full-system baseline, reaches 44/70ms; Qdrant reaches 68/101ms.

4.4 Nearest Systems and Public Replication

Table 2 separates execution authorization from backend I/O. The Gan–Wang planner reaches 48/76ms, and GateANN reaches 44/70ms; their five-run intervals are above RAVEL's 37/56ms. GateANN uses 534 SSD reads/query versus 618 for RAVEL, confirming its backend I/O advantage. Pair-conditioned authorization improves the latency tail without replacing the index.

On the 356 public final-test band queries, RAVEL records 38.9/59.7ms P95/P99 with 0.958 Recall@20. The corresponding point estimates are 41.8/64.6ms for RCPS-Tail and 58.7/91.4ms for the workload's unmodified native plan selector; marginal intervals in Table 3 quantify run variation. In the 341 low- and 327 high-selectivity queries, RAVEL remains within 2.8ms of the best method.

4.5 Quality, Calibration, and Systems Behavior

Under the separate identical-80ms protocol, RAVEL sustains 0.94–0.98 Recall@20 across temporal, categorical, hierarchical, and multi-constraint requests (Appendix Table 11). The exact-oracle gap

Table 2: Closest-system comparison on Medium+EntDoc failure-band queries under identical CPU, RAM, SSD, cache, and a common 0.95 Recall@20 target. Achieved recall is reported per row; brackets give 95% run-level bootstrap intervals over five independent runs.

System	P95 (ms)	P99 (ms)	Recall@20	SSD reads/query
Qdrant	68 [65.2,71.4]	101 [95.0,108.2]	0.956	847
Learned planner (Gan & Wang, 2026)	48 [45.8,50.5]	76 [71.5,81.3]	0.956	805
GateANN (Lee et al., 2026)	44 [41.9,46.4]	70 [65.9,74.8]	0.953	534
RAVEL	37 [35.9,38.4]	56 [53.4,59.6]	0.958	618

Table 3: Public MoReVec-style replication in the frozen 5–15% band. The 356 final-test queries use the enterprise study's P1–P4 definitions, exact oracle, common 0.95 Recall@20 target, and five-run protocol. Workload-specific quantile and RCPS-Tail corrections use a template-disjoint public calibration partition; RAVEL's gate thresholds transfer unchanged. Brackets give marginal 95% run-level bootstrap intervals.

Method	P95 (ms)	P99 (ms)	Recall@20
Native policy	58.7 [55.4,62.3]	91.4 [85.8,98.1]	0.952
Winner-only	47.9 [45.2,50.9]	74.8 [69.8,80.6]	0.956
Point margin	43.6 [41.1,46.4]	68.5 [64.0,73.5]	0.957
RCPS-Tail	41.8 [39.5,44.4]	64.6 [60.2,69.8]	0.958
RAVEL	38.9 [36.8,41.2]	59.7 [55.6,64.4]	0.958

remains below 0.02 after fanout reaches 100. This complements, but is not pooled with, the primary common-target latency results.

Removing P3, fallback, or learned ranking raises failure-band P95 by 24–51% (Appendix Table 8). Among same-action controllers, point-margin, width-only, and RCPS-Tail successively reach 42, 40, and 39ms P95, while pair-conditioned authorization reaches 37ms. On final-test requests, the frozen provisional-action upper quantile improves empirical coverage from 0.902 to 0.949 overall and reaches 0.944 in the band (Appendix Table 6). Post-selection sweeps retain a useful neighborhood around the selected overlap, uncertainty, and safe-mode thresholds (Appendix Table 7).

The controller adds only 0.18ms online overhead. At $\pm 20\%$ selectivity noise, mis-picks remain 5.1% and P95 grows 3.6%; at $\pm 40\%$, both remain below 8% and 6%, while $\pm 30\%$ correlation noise gives 5.9% mis-picks and 4.1% P95 growth. Appendix Figure 6 breaks down this control cost. The 85M-vector configuration occupies 94GB RAM and 420GB SSD, versus 153GB RAM for all-HNSW at comparable recall; Figure 5 distinguishes measured deployment markers from model-predicted sweep context.

On LegalDoc, 5K calibration reaches 40.9ms P95 versus 40.1ms in-domain, and hierarchy-aligned P4 reaches 39.1ms versus 45.5ms for P3. RAVEL also retains lower tails than GateANN from 10M to 100M single-index settings and at four-shard 200M aggregate capacity (Appendix Tables 13–15).

Discussion and scope. The evidence establishes pair-conditioned authorization for recurring CPU-only retrieval under explicit recall, latency, RAM, and SSD caps. Common-target latency and fixed-deadline recall remain separate estimands; cold starts use a feasible P3 path, abrupt shifts activate residual-driven safe mode, and offline oracle labels never enter serving decisions. The controlled replay measures per-request tails, while saturated multi-query throughput and generation quality are end-to-end system properties beyond this retrieval-layer estimand.

5 Conclusion

Pair-conditioned authorization adds execution evidence to a planner's top-1 rank. With actions and fallback fixed, RAVEL cuts executed-policy mis-picks from 12.6% to 4.9%, reaches 37/56ms P95/P99 at the common recall target, and lowers selective P95 excess over RCPS-Tail from 14.6ms to 9.6ms. It preserves retrieval budget without replacing the index.

AI USE STATEMENT

Generative AI tools assisted manuscript preparation through language editing, formatting checks, and bibliographic consistency review. The authors reviewed all AI-assisted edits and take responsibility for the final content.

REFERENCES

Abylay Amanbayev, Brian Tsan, Tri Dang, and Florin Rusu. Filtered approximate nearest neighbor search in vector databases: System design and performance analysis. *arXiv preprint arXiv:2602.11443*, 2026. doi: 10.48550/arXiv.2602.11443. URL https://arxiv.org/abs/2602.11443.

Yuzheng Cai, Jiayang Shi, Yizhuo Chen, and Weiguo Zheng. Navigating labels and vectors: A unified approach to filtered approximate nearest neighbor search. *Proceedings of the ACM on Management of Data*, 2(6):246:1–246:27, 2024. doi: 10.1145/3698822. URL https://doi.org/10.1145/3698822.

Qi Chen, Bing Zhao, Haidong Wang, Mingqin Li, Chuanjie Liu, Zengzhong Li, Mao Yang, and Jingdong Wang. SPANN: Highly-efficient billion-scale approximate nearest neighbor search. In *Advances in Neural Information Processing Systems*, volume 34, pp. 5199–5212, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/299dc35e747eb77177d9cea10a802da2-Abstract.html.

Zeheng Fan, Yuxiang Zeng, Zhuanglin Zheng, and Yongxin Tong. FedVSE: A privacy-preserving and efficient vector search engine for federated databases. *Proceedings of the VLDB Endowment*, 18(12):5371–5374, 2025. doi: 10.14778/3750601.3750674. URL https://www.vldb.org/pvldb/vol18/p5371-tong.pdf.

Zhuocheng Gan and Yifan Wang. Efficient Filtered-ANN via learning-based query planning. *arXiv preprint arXiv:2602.17914*, 2026. doi: 10.48550/arXiv.2602.17914. URL https://arxiv.org/abs/2602.17914.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *CoRR*, abs/2312.10997, 2023. doi: 10.48550/arXiv.2312.10997. URL https://arxiv.org/abs/2312.10997.

Siddharth Gollapudi, Neel Karia, Varun Sivashankar, Ravishankar Krishnaswamy, Nikit Begwani, Swapnil Raz, Yiyong Lin, Yin Zhang, Neelam Mahapatro, Premkumar Srinivasan, Amit Singh, and Harsha Vardhan Simhadri. Filtered-DiskANN: Graph algorithms for approximate nearest neighbor search with filters. In *Proceedings of the ACM Web Conference 2023*, pp. 3406–3416. ACM, 2023. doi: 10.1145/3543507.3583552. URL https://doi.org/10.1145/3543507.3583552.

Roman Heinrich, Xiao Li, Manisha Luthra, and Zoi Kaoudi. Learned cost models for query optimization: From batch to streaming systems. *Proceedings of the VLDB Endowment*, 18(12):5482–5487, 2025a. doi: 10.14778/3750601.3750699. URL http://dx.doi.org/10.14778/3750601.3750699.

Roman Heinrich, Manisha Luthra, Johannes Wehrstein, Harald Kornmayer, and Carsten Binnig. How good are learned cost models, really? insights from query optimization tasks. *Proceedings of the ACM on Management of Data*, 3(3):172:1–172:27, 2025b. doi: 10.1145/3725309. URL https://doi.org/10.1145/3725309.

Guoyu Hu, Shaofeng Cai, Tien Tuan Anh Dinh, Zhongle Xie, Cong Yue, Gang Chen, and Beng Chin Ooi. HAKES: Scalable vector database for embedding search service. *Proceedings of the VLDB Endowment*, 18(9):3049–3062, 2025. doi: 10.14778/3746405.3746427. URL https://www.vldb.org/pvldb/vol18/p3049-ooi.pdf.

Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2011. doi: 10.1109/TPAMI.2010.57.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 6769–6781. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550/.

Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 39–48. Association for Computing Machinery, 2020. doi: 10.1145/3397271.3401075. URL https://doi.org/10.1145/3397271.3401075.

Kyoungmin Kim, Jisung Jung, In Seo, Wook-Shin Han, Kangwoo Choi, and Jaehyok Chong. Learned cardinality estimation: An in-depth study. In *International Conference on Management of Data (SIGMOD '22)*, pp. 1214–1227. ACM, 2022. doi: 10.1145/3514221.3526154. URL https://doi.org/10.1145/3514221.3526154.

Nakyung Lee, Soobin Cho, Jiwoong Park, and Gyuyeong Kim. GateANN: I/O-efficient filtered vector search on SSDs. *arXiv preprint arXiv:2603.21466*, 2026. doi: 10.48550/arXiv.2603.21466. URL https://arxiv.org/abs/2603.21466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020. doi: 10.48550/arXiv.2005.11401. URL https://arxiv.org/abs/2005.11401.

Zhaoheng Li, Silu Huang, Wei Ding, Yongjoo Park, and Jianjun Chen. SIEVE: Effective filtered vector search with collection of indexes. *Proceedings of the VLDB Endowment*, 18(11):4723–4736, 2025. doi: 10.14778/3749646.3749725. URL http://dx.doi.org/10.14778/3749646.3749725.

Anqi Liang, Pengcheng Zhang, Bin Yao, Zhongpu Chen, Yitong Song, and Guangxu Cheng. UNIFY: Unified index for range filtered approximate nearest neighbors search. *Proceedings of the VLDB Endowment*, 18(4):1118–1130, 2024. doi: 10.14778/3717755.3717770. URL https://www.vldb.org/pvldb/vol18/p1118-yao.pdf.

Yanjun Lin, Kai Zhang, Zhenying He, Yinan Jing, and X. Sean Wang. Survey of filtered approximate nearest neighbor search over the vector-scalar hybrid data. *arXiv preprint arXiv:2505.06501*, 2025. doi: 10.48550/arXiv.2505.06501. URL https://arxiv.org/abs/2505.06501.

Hanwen Liu, Shashank Giridhara, and Ibrahim Sabek. Conformal prediction for verifiable learned query optimization. *Proceedings of the VLDB Endowment*, 18(8):2653–2666, 2025. doi: 10.14778/3742728.3742755. URL http://dx.doi.org/10.14778/3742728.3742755.

Jiarui Luo, Miao Qiao, Chaoji Zuo, and Dong Deng. Tag-filtered approximate nearest neighbor search. In *41st IEEE International Conference on Data Engineering (ICDE 2025)*, pp. 3642–3654. IEEE, 2025. doi: 10.1109/ICDE65448.2025.00272. URL https://doi.org/10.1109/ICDE65448.2025.00272.

Yu A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836, 2020. doi: 10.1109/TPAMI.2018.2889473.

Ryan Marcus, Parimarjan Negi, Hongzi Mao, Nesime Tatbul, Mohammad Alizadeh, and Tim Kraska. Bao: Making learned query optimization practical. In *Proceedings of the 2021 International Conference on Management of Data (SIGMOD '21)*, pp. 1275–1288. ACM, 2021. doi: 10.1145/3448016.3452838. URL http://dx.doi.org/10.1145/3448016.3452838.

Liana Patel, Peter Kraft, Carlos Guestrin, and Matei Zaharia. ACORN: Performant and predicate-agnostic search over vector embeddings and structured data. *Proceedings of the ACM on Management of Data*, 2(3):120:1–120:27, 2024. doi: 10.1145/3654923. URL http://dx.doi.org/10.1145/3654923.

Ji Sun, Jintao Zhang, Zhaoyan Sun, Guoliang Li, and Nan Tang. Learned cardinality estimation: A design space exploration and a comparative evaluation. *Proceedings of the VLDB Endowment*, 15 (1):85–97, 2021. doi: 10.14778/3485450.3485459. URL http://www.vldb.org/pvldb/vol15/p85-li.pdf.

Ji Sun, Guoliang Li, James Pan, Jiang Wang, Yongqing Xie, Ruicheng Liu, and Wen Nie. GaussDB-Vector: A large-scale persistent real-time vector database for LLM applications. *Proceedings of the VLDB Endowment*, 18(12):4951–4963, 2025. doi: 10.14778/3750601.3750619. URL http://dx.doi.org/10.14778/3750601.3750619.

Saravanan Thirumuruganathan, Suraj Shetiya, Nick Koudas, and Gautam Das. Prediction intervals for learned cardinality estimation: An experimental evaluation. In *38th IEEE International Conference on Data Engineering (ICDE 2022)*, pp. 3051–3064. IEEE, 2022. doi: 10.1109/ICDE53745.2022.00274. URL https://doi.org/10.1109/ICDE53745.2022.00274.

Zilong Wang, Qixiong Zeng, Ning Wang, Haowen Lu, and Yue Zhang. Ceda: Learned cardinality estimation with domain adaptation. *Proceedings of the VLDB Endowment*, 16(12):3934–3937, 2023. doi: 10.14778/3611540.3611589. URL https://www.vldb.org/pvldb/vol16/p3934-wang.pdf.

Qian Xu, Feng Zhang, Chengxi Li, Lei Cao, Zheng Chen, Jidong Zhai, and Xiaoyong Du. HARMONY: A scalable distributed vector database for high-throughput approximate nearest neighbor search. *Proceedings of the ACM on Management of Data*, 3(4):249:1–249:28, 2025. doi: 10.1145/3749167. URL https://doi.org/10.1145/3749167.

Fangyuan Zhang, Mengxu Jiang, Guanhao Hou, Jieming Shi, Hua Fan, Wenchao Zhou, Feifei Li, and Sibo Wang. Efficient dynamic indexing for range filtered approximate nearest neighbor search. *Proceedings of the ACM on Management of Data*, 3(3):152:1–152:26, 2025a. doi: 10.1145/3725401. URL https://doi.org/10.1145/3725401.

Jintao Zhang, Chao Zhang, Guoliang Li, and Chengliang Chai. Autoce: An accurate and efficient model advisor for learned cardinality estimation. In *39th IEEE International Conference on Data Engineering (ICDE 2023)*, pp. 2621–2633. IEEE, 2023. doi: 10.1109/ICDE55515.2023.00201. URL https://doi.org/10.1109/ICDE55515.2023.00201.

Jintao Zhang, Chao Zhang, Guoliang Li, and Chengliang Chai. Pace: Poisoning attacks on learned cardinality estimation. *Proceedings of the ACM on Management of Data*, 2(1):37:1–37:16, 2024. doi: 10.1145/3639292. URL https://doi.org/10.1145/3639292.

Qihan Zhang, Shaolin Xie, and Ibrahim Sabek. Limao: A framework for lifelong modular learned query optimization. *Proceedings of the VLDB Endowment*, 18(11):4546–4559, 2025b. doi: 10.14778/3749646.3749712. URL https://www.vldb.org/pvldb/vol18/p4546-zhang.pdf.

Bolong Zheng, Ziyang Yue, Qi Hu, Xiaomeng Yi, Xiaofan Luan, Charles Xie, Xiaofang Zhou, and Christian S. Jensen. Learned probing cardinality estimation for high-dimensional approximate nn search. In *39th IEEE International Conference on Data Engineering (ICDE 2023)*, pp. 3209–3221. IEEE, 2023. doi: 10.1109/ICDE55515.2023.00246. URL https://doi.org/10.1109/ICDE55515.2023.00246.

Chaoji Zuo, Miao Qiao, Wenchao Zhou, Feifei Li, and Dong Deng. SeRF: Segment graph for range-filtering approximate nearest neighbor search. *Proceedings of the ACM on Management of Data*, 2(1):69:1–69:26, 2024. doi: 10.1145/3639324. URL http://dx.doi.org/10.1145/3639324.

A Controller and Evaluation Details

A.1 Runtime Interfaces and Audit State

The runtime path separates request normalization, recall eligibility, tail prediction, authorization, and execution. Table 4 gives the component interfaces; Table 5 then makes each state transition and cold-start response explicit.

Table 4: Interfaces among RAVEL's frontend, estimator, planner, and execution engine. Uncertainty and authorization outcomes remain explicit at component boundaries.

Block	Input	Output
Frontend	(q, Π, SLO)	normalized spec
Estimator	spec, sketches	$(\hat{s}, \hat{r}, \hat{u})$
Planner	features, budgets	plan p, fanout k'
Engine	p, k', resource caps	results, telemetry

Table 5: Auditable state transitions for budget-conditioned authorization. Every veto and safe-mode transition is reconstructable from the listed telemetry.

Stage	Observable / key	Fixed rule / constant	Action if violated or triggered
Recall-floor lookup	dataset, hardware, workload family, requested k, selectivity bin, correlation bin	choose the smallest k' whose held-out replay meets ρ=0.95; sparse cells back off to a coarser key	candidate is not compared until a recall floor is available
Budget filter	$\widehat{R}$, $\widehat{M}$, $\widehat{D}$, $\widehat{L}_{95}$, $(\rho, m_{\max}, d_{\max}, \tau)$	retain candidates meeting the recall floor and RAM, SSD, and latency caps	infeasible candidates are dropped before ranking
Winner proposal	$\widehat{L}_{50}$ and $\widehat{L}_{95}$ for each (p, k')	provisional winner is the minimum-$\widehat{L}_{95}$ feasible candidate	top-2 pair is passed to the tail-separation gate
Tail-separation gate	overlap $\mathrm{ov}(a, b)$, summary uncertainty $\hat{u}$	thresholds selected on gate validation; veto if $\mathrm{ov}(a, b) > 0.15$ or $\hat{u} > 0.3$	reuse the safest recent feasible plan for the same regime key
Fallback cache	regime key, held-out recall eligibility, realized P95, SSD probes	valid recall-table key first, then lower realized P95, then fewer SSD probes; ties break toward P3	use feasible P3 at cold start, else lowest-P95 feasible plan; record cold-start flag
Safe mode	1-minute realized P95 residual-violation rate	enter above 15%; exit below 5%	route failure-band queries to P3 and use threshold heuristics elsewhere
Retrain trigger	predicted/observed P95 divergence on latest 10K queries	retrain asynchronously on the most recent 50K telemetry records	keep previous checkpoint for rollback until the new model is accepted

Calibration and gate selection use different template groups. Table 6 evaluates the frozen quantile correction on final-test requests, Table 7 varies already-selected thresholds only as a robustness diagnostic, and Table 8 removes execution or control components.

Table 6: Final-test audit of Medium+EntDoc provisional-winner tail quantiles. The correction is fit on a template-disjoint calibration partition and frozen before evaluating one provisional action per reported request (3,248 total; 812 in-band). The lower panel stratifies those request-level checks by provisional action.

Model	All coverage	Band coverage	Mean span (ms)	N_{req} (all/band)
Uncalibrated	0.902	0.884	10.9	3,248/812
Frozen calibrated	0.949	0.944	12.7	3,248/812

Post-calibration plan	P1	P2	P3	P4
Empirical coverage	0.947	0.951	0.943	0.948

The calibrated provisional-action upper quantile reaches its intended marginal coverage overall and remains close in the failure band. Pairwise decision quality is measured separately by Figure 3.

Table 7: Post-selection one-factor sensitivity on final-test Medium+EntDoc band queries. Actions, checkpoint, recall tables, requests, budgets, and fallback implementation remain fixed; the gate-validation operating point is θ=0.15, $\sigma_{\max}$=0.30, with 15%/5% residual safe-mode entry/exit. Mis-pick is the executed-policy metric.

Family	Setting	P95	P99	Exec. mis-pick	Fallback
θ	0.10	38.2	58.6	5.3%	21.4%
	0.15	37	56	4.9%	18.3%
	0.20	38.0	58.8	5.5%	14.0%
$\sigma_{\max}$	0.25	38.4	59.1	5.4%	22.2%
	0.30	37	56	4.9%	18.3%
	0.35	39.1	60.5	5.8%	13.8%
Safe mode	10%/3%	38.3	60.1	–	–
	15%/5%	37	56	–	–
	20%/8%	39.5	62.7	–	–

No single threshold perturbation reverses the reported tail pattern. The fallback ablation below is larger because it removes the alternative execution path rather than moving its trigger.

Table 8: Failure-band ablation on Medium+EntDoc. Execution kernels are unchanged; rows remove controller components or restrict which established actions are available. Table 1 separately holds the complete P1–P4 menu fixed.

Variant	P95 (ms)	Δ	Note
Full (P1–P4 + fallback)	37	–	reference
minus P3	46	+24%	no interleaving
minus fallback	50	+35%	risky mis-picks
P1+P2 only	52	+41%	binary limit
Heuristic (no learned model)	56	+51%	threshold-based

A.2 Positioning and Baseline Configuration

Table 9 distinguishes changes to an index, a plan ranker, and an authorization rule. Table 10 records the settings that determine the common-target recall and matched-coverage comparisons.

Table 9: Decision-level positioning relative to adjacent filtered-search research. The comparable-evidence column points to direct measurements in this paper.

System	Decision object	Budget / uncertainty treatment	Comparable evidence	Operational consequence
Gan & Wang (2026)	top-ranked filtered-ANN plan	learned cost ranking under query features	Table 2	efficient winner prediction
GateANN (Lee et al., 2026)	SSD traversal and layout	backend-local I/O control	Table 2	fewer cold-tier reads
Amanbayev et al. (2026)	selectivity/correlation operating region	workload-conditioned system analysis	Tables 3 and 12	reproducible stress geometry
Practical systems	deployed fixed/backend policies	tuned to a common recall target and resources	Table 2	strong serving references
RAVEL	authorization of a resource-feasible action	calibrated marginal spans plus pair score	Table 1; Figure 3	executes separated winners

Table 10: Decision-critical baseline configurations under matched CPU-only budgets and a common ρ=0.95 recall-table target. Fanout is selected before final testing.

System	Index	Filter Exec.	Key Params
FF	HNSW/IVF	pre-filter	M=32, ef_s=128
AF	HNSW	post-filter	k'=200, expand-on-miss
Milvus	IVF_FLAT	bitmap + scan	n_{list}=4096
Qdrant	HNSW	payload filter	M=32, ef_s=128
ACORN	IVF	filter-aware routing	n_{list}=4096, 2-stage probe
SIEVE	H.Quant.	predicate hints	3-level quantizer, cache-aware
Learned planner (Gan & Wang, 2026)	P1–P4	winner dispatch	learned latency rank; common recall table
GateANN (Lee et al., 2026)	SSD graph	gated traversal	common recall target; tuned fanout
RCPS-Tail	P1–P4	selective fallback	paired excess target; one-sided 0.95 correction
RAVEL	P1–P4	pair-conditioned fallback	P50/P95 LightGBM; θ=0.15, $\sigma_{\max}$=0.30

A.3 Quality, Public Counts, and Transfer

The first table uses the fixed-80ms protocol. The public and scale comparisons use the common ρ=0.95 recall-table target and report achieved recall; transfer and hierarchy tables instead report condition-specific recall under their matched-budget diagnostics. These estimands are interpreted separately.

Table 11: Recall@20 on Medium+EntDoc under an identical 80ms latency budget. Unlike Table 2, this table fixes latency and permits recall to vary. The oracle is exact brute force.

Pattern (Sel.)	FF	AF	Qdrant	ACORN	SIEVE	RAVEL
Temp. (0.5%)	0.971	0.818	0.948	0.962	0.954	0.978
Cat. (5%)	0.938	0.782	0.912	0.928	0.919	0.958
Hier. (12%)	0.891	0.847	0.886	0.912	0.905	0.943
Multi (2%)	0.922	0.811	0.901	0.918	0.908	0.962

Table 12: Public final-test counts by selectivity and vector–metadata correlation. The 1,024 queries are template-disjoint from the public calibration partition.

Selectivity	Low corr.	Moderate corr.	High corr.	Total
$< 5\%$	91	173	77	341
5–15%	68	211	77	356
$> 15\%$	62	149	116	327

The public final-test population covers all three selectivity ranges and correlation levels. Its band denominator is used only for Table 3; low- and high-selectivity bins support the outside-band comparison in Section 4.4.

Table 13: Quantitative transfer on the held-out LegalDoc failure band. Recalibration progressively approaches the in-domain controller; mis-pick is evaluated on the executed policy against the resource-feasible paired-trace oracle.

Condition	P95	P99	Recall@20	Exec. mis-pick
Best fixed baseline	73.8	110.6	0.907	–
In-domain RAVEL	40.1	62.3	0.953	4.6%
Zero-shot	57.9	88.7	0.932	11.9%
+1K calibration	49.0	74.8	0.945	7.7%
+5K calibration	40.9	63.8	0.950	5.1%

Recalibration closes most of the LegalDoc zero-shot gap by 5K requests without changing the action definitions. The hierarchy-aligned action comparison below separates that transfer effect from the benefit of P4's partition geometry.

Table 14: Action-level results on hierarchy-aligned LegalDoc queries. P4 benefits when predicate taxonomy and partition geometry coincide, while P3 remains the strongest general fallback.

Action	P95	P99	Recall@20	SSD reads/query
P1 filter-first	65.2	99.1	0.914	843
P2 ANN-first	80.7	125.8	0.889	1294
P3 interleaved	45.5	70.4	0.947	682
P4 partition-aware	39.1	61.8	0.955	527

P4 is strongest on this taxonomy-aligned slice, while P3 remains the general cold-start fallback because it does not require partition alignment. The scale matrix then compares RAVEL and GateANN under each configuration's explicit RAM and SSD caps.

Table 15: Scale and deployment matrix under a common 0.95 Recall@20 target and per-configuration resource caps. Each setting uses final-test band queries and five independent runs; brackets are 95% run-level bootstrap intervals, RAM/SSD report peak aggregate GB, and reads are per query. The 10M setting is a document-ID-stable LegalDoc subset; 200M is aggregate capacity across four disjoint 50M EntDoc tenant shards, with each query routed to one shard.

Setting	System	P95 [CI]	P99 [CI]	R@20	RAM/SSD (GB)	Reads
Small+LegalDoc, 10M	GateANN	46.3 [43.9,48.9]	71.8 [67.4,76.9]	0.951	27.8/91.4	421
	RAVEL	36.2 [34.5,38.1]	55.7 [52.3,59.6]	0.957	28.6/88.7	468
Medium+EntDoc, 50M	GateANN	44 [41.9,46.4]	70 [65.9,74.8]	0.953	107.5/438	534
	RAVEL	37 [35.9,38.4]	56 [53.4,59.6]	0.958	109.4/420	618
Medium+EcomProd, 100M	GateANN	58.4 [55.2,61.9]	91.2 [85.5,97.6]	0.949	119.3/812	702
	RAVEL	44.6 [42.2,47.3]	68.8 [64.5,73.7]	0.955	117.8/786	768
Cluster+EntDoc, $4\times$50M	GateANN	54.1 [51.2,57.3]	84.6 [79.4,90.5]	0.948	176.2/1752	812
	RAVEL	41.8 [39.6,44.2]	64.5 [60.6,68.9]	0.954	173.9/1676	876

The 200M row is aggregate service capacity across four independently routed 50M shards, not a 200M-vector single-query fanout. Table 16 closes the supplement by matching each controlled resource to its measured stressor and comparator.

Table 16: Evidence matrix for the planner–engine authorization boundary. Each protected serving object is paired with its principal stressor, enforced property, nearest comparator, evaluation setting, utility cost, and manuscript evidence.

Protected object	Stressor	Enforced property	Comparator	Setting	Utility / cost	Evidence trace
Latency SLO	action-rank reversal	authorize only after feasible-pair separation	winner-only; RCPS-Tail	template-held-out failure band	37/56ms P95/P99	Eq. 1; Fig. 3; Tables 1, 2
Recall floor	low fanout	admit candidates meeting ρ before ranking	fixed fanout	held-out exact-oracle table	0.953–0.958 main-band recall	Algorithm 1; Table 2
RAM/SSD caps	hot-set and cold-index expansion	remove actions above $(m_{\max}, d_{\max})$ before authorization	all-HNSW	20M–200M deployments	94/420 GB at 85M vectors	Eq. 1; Fig. 5; Table 15
SSD exposure	random cold-tier probes	veto confusable high-I/O actions	GateANN	final-test band; common recall target	0.18ms control overhead	Tables 2, 1; Fig. 6
Feature integrity	selectivity or correlation noise	fallback limits decision-error propagation	RCPS-Tail	20% and 40% perturbations	under 6% P95 loss at 40%	Fig. 3; Table 7
Model state	abrupt workload drift	realized residuals activate safe mode	static checkpoint	threshold perturbation study	fixed 15%/5% trigger	Algorithm 1; Table 7

A.4 Interleaved Execution Kernel

Time, memory, fanout, and frontier exhaustion are the only stopping conditions. An incomplete result remains in recall accounting and is emitted with the budget that stopped traversal; it is not silently padded or removed from evaluation.

Algorithm 2 P3 interleaves bounded ANN traversal with predicate evaluation. Predicate failures avoid result-set insertion but do not suppress neighbor expansion, preserving graph reachability until an explicit budget terminates traversal.

Require: query q, predicate Π, budgets $(t_{\max}, m_{\max})$, target k, action fanout k'
Initialize frontier H, visited set V, result set R (size k), counters $M \leftarrow 0$, $e \leftarrow 0$
Push the backend's seed entry into H with its native ANN priority
while time$< t_{\max}$ **and** $M < m_{\max}$ **and** $e < k'$ **and** $H \neq \emptyset$ **do**
  $(u, \widehat{d}) \leftarrow H$.pop()
  if $u \in V$ **then**
    continue
  end if
  V.insert(u); $e \leftarrow e + 1$; $M \leftarrow M + c_{\text{state}}$
  if $\Pi(u)$ **then**
    $d \leftarrow \text{dist}(q, u)$; R.maybeInsert(u, d)
  end if
  for each neighbor $v \in N(u)$ **do**
    if $v \in V$ **then**
      continue
    end if
    H.push$(v, \text{ANNPriority}(q, v))$
  end for
end while
status $\leftarrow$ COMPLETE if $|R| = k$, else INCOMPLETE
return $(R, \textit{status})$

B SUPPLEMENTARY FIGURES

Figure 4 locates the measured enterprise workload over the two stress-test axes. The central density intersects the same 5–15% transition region isolated by Figure 1.

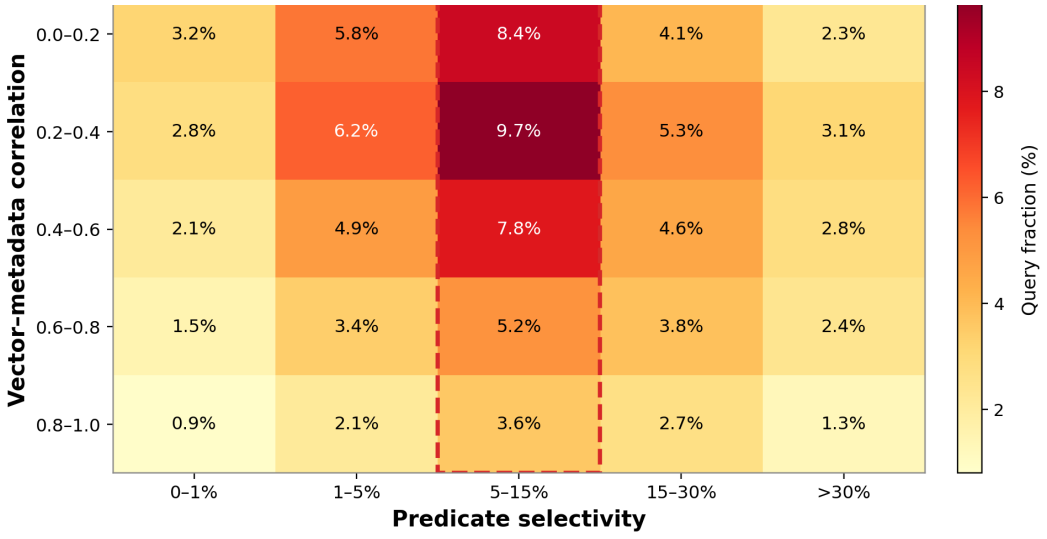


Figure 4: Descriptive enterprise trace before template grouping and final-test assignment. The 5–15% column contains 34.7% of this raw mixture; the distinct final-test cohort contains 812/3,248 (25.0%) band requests.

Figure 5 places directly measured deployment markers over model-predicted sweep context. The marker boundary shows how tier-aware authorization converts fixed RAM and SSD capacity into additional feasible operating points.

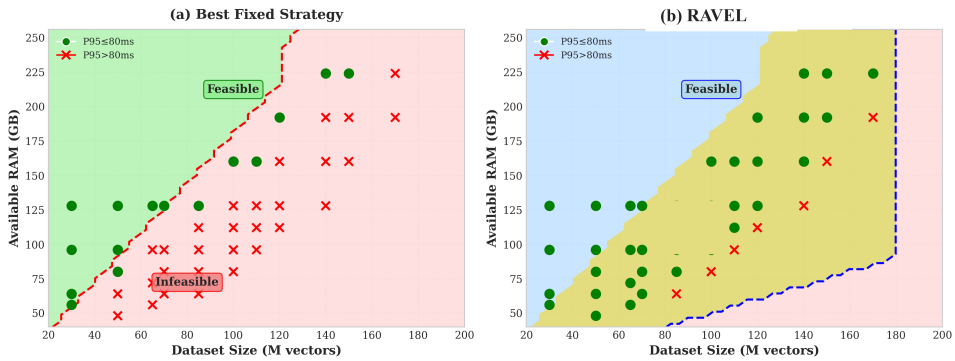
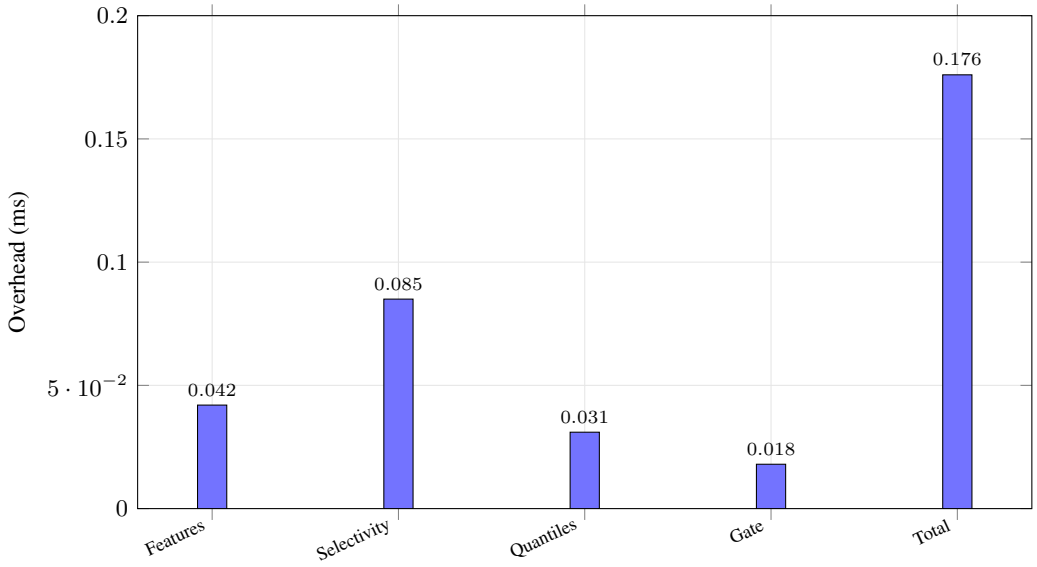


Figure 5: Budget feasibility under P95≤80ms and a 1TB SSD budget. Circles/crosses are measured pass/fail points. Background colors are model predictions: green/blue denotes fixed-policy feasibility, gold denotes capacity made feasible by RAVEL, and pink denotes infeasibility for both.



Headline-band outcome	Authorized	Fallback
Share of 4,060 query–run observations	81.7%	18.3%

Figure 6: Online control cost and decision outcomes on Medium+EntDoc. Feature extraction, selectivity estimation, quantile scoring, and gate evaluation total 0.176ms (reported as 0.18ms); the fallback denominator matches Table 1.

Figure 6 decomposes online control cost and reports denominator-matched authorization and fallback outcomes. Total overhead remains 0.18ms.

The 0.176ms total is less than 0.5% of the 37ms headline P95, while the 18.3% fallback share uses the same 4,060 query–run denominator as Table 1.